

Jonathan Shidal

CSE 332S/504N

CSE 332S/504N: Object-oriented software development laboratory

All course materials such as lecture slides, studios, and labs will be assigned and distributed via [Canvas](#).

All questions should be directed to the course discussion board via [Piazza](#).

Description

This course will expose you to the syntax, semantics, and standard libraries of the C++ language; to common C++ design and implementation idioms; to object-oriented design; and to many opportunities to apply the knowledge and idioms you learn through hands-on studio exercises and projects in C++.

All programming in this course will be done in C++, a language in which you can combine the object-oriented, procedural, and generic programming ideas with which you'll gain experience throughout the semester. Since many students enter this course with experience in Java but not C++, the early portion of the syllabus is designed to give you a breadth-first experience of the major capabilities and features of C++ and to assist your progress up the rest of the C++ learning curve. The final third of this course will focus on object-oriented programming principles and design.

Meeting times and location

Section 01: Mondays and Wednesdays, 10-11:20 AM in EADS 016

Section 02: Tuesdays and Thursdays, 10-11:20 AM in EADS 016

Section 03: Mondays and Wednesdays, 1-2:20 PM in EADS 016

Instructor

Jon Shidal

Office: Jolley Hall 214

Email: shidalj@wustl.edu

Office hours: TBD

Prerequisites

CSE247, Data Structures and Algorithms (possible to take concurrently with CSE 332, but please ask)

Proficiency with Java (e.g., from CSE132), C, C++ or other general-purpose industrial-strength programming language.

Textbooks (both available free online via WUSTL libraries)

[LLM] Stanley B. Lippman, Jose Lajoie, and Barbara E. Moo, C++ Primer, Fifth Edition, ISBN 0321714113 , Prentice Hall, 2012.

[GHJV] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
Design Patterns: Elements of Reusable Object-Oriented Software,
ISBN 0321700694

Discussion board

We will be using [Piazza](#) as our course discussion board. Discussion of studios, practice exam problems, lecture slides, and other course content is encouraged and can help ensure you are understanding concepts properly. If you have a question, it is likely one of your classmates does as well. Please ask general questions covering course concepts and studios directly to Piazza so that your classmates may benefit from your question as well. **Lab specific questions** that require sharing code or any other aspects of your solution should be asked directly to an instructor (via a private Piazza post or email) or TA during office hours.

The sign-up link for the course Piazza page is [here](#).

Studios

Each lecture is accompanied by a studio. Each studio consists of required exercises and possibly some optional exercises. The required exercises for each studio must be completed in full to receive full participation credit in the course. Each student will have access to a repository managed via Github that will be used to submit studio work. Please refer to studio documents for details on submission.

Labs

There will be 5 lab assignments in this course. The labs will increase in difficulty and value throughout the semester.

Labs are to be submitted via Github, and will be graded and returned to you via Github as well. Each lab submission requires a separate `ReadMe.txt` file containing a lab report write-up. Each assignment document will describe the procedure for creating and cloning a Github repository to hold your code, along with the report format and content requirements. Use of this process is required.

Labs 1, 2, and 3 are to be completed individually, though on labs 4 and 5 you will have the option of working in teams of 2 or 3 people.

Late policy:

You should submit your code and a lab report for each lab ahead of its advertised due date and time.

Labs received within 24 hours after the posted deadline will be accepted with a 10% penalty up front, and labs submitted between 24 and 48 hours after the posted deadline will be accepted with a 20% penalty up front. Labs submitted after that will not be graded, except in the case of **documented** extenuating circumstances.

Extensions may be granted in advance on a case-by-case basis as long as the circumstances warrant an extension and the request is made sufficiently ahead of the deadline. Extensions

will not be granted the day of or after the deadline except in very special cases (such as documented medical emergencies).

You may discuss general questions related to your projects with other students and your instructor during studio hours. For additional help please contact your instructor. While you are encouraged to ask for help on particular problems you may encounter during a project, you must design and implement your own solutions, and prepare your own reports. In particular, while discussion of small code fragments related to specific questions is allowed and encouraged, direct sharing, viewing, or other “reuse” of complete code files (especially if done covertly) is strictly forbidden. If you’re not sure whether or not sharing a given section of code is ok, please ask your instructors’ permission before doing so.

Programming Guidelines:

Many organizations impose coding standards on programmers, and these standards can range from general principles to very specific requirements. In this class, part of the grade for each programming assignment will be based on the [CSE 332 Programming Guidelines](#), which are intended to build good programming habits that will help avoid common mistakes and help make your programs more readable and better organized and documented. As the course progresses, we’ll point out relevant ones for each programming assignment, so please re-review it before starting each new programming assignment.

Academic integrity

Cheating is the misrepresentation of someone else’s work as your own, or assisting someone else in cheating, and will not be tolerated in this course.

For this course, examples of cheating include but are not limited to:

- providing or receiving answers or other assistance during an exam,
- accessing unapproved sources of information during an exam,
- submitting program code written outside this course this semester or by another person not on your team, without specific prior approval by the instructors,
- allowing another student not on your team to copy (or helping them reconstruct) essentially the same solution your team has developed for a programming assignment.

This is a very serious matter. Anyone found cheating will at a minimum **lose** points equal to the assigned value for the assignment in question (for example if an assignment were worth 10% of the course grade then -10% of the course grade would be assigned), or possibly receive an F for the course. Further action may be taken in extreme cases.

Furthermore, our policy is that we will make the final determination on what constitutes cheating. If you suspect that you may be entering an ambiguous situation, it is your responsibility to clarify it before we detect it. If in doubt, please ask.

To ensure this policy is implemented consistently and fairly, and that any problems are detected and addressed as early as possible, we may make use of automated program similarity checkers to assess uniqueness and independence of submitted program solutions.

Grading breakdown

The final grade for this course follows the breakdown below.

Assignment	% of final grade
Lab 1	8%
Lab 2	10%
Lab 3	12%
Lab 4	15%
Lab 5	20%
Studios and Participation	15%
Exam	19%
Course eval.	1%

Letter grades will be assigned based on the typical 10 point grading scale. 90-100 is an A, 80-89 is a B, and so forth. + and – will be added within 2 points of a boundary. For example, 80-81 will be a B-, 82-87 will be a B, and 88-89 will be a B+.

For students taking the course pass/fail, a grade of a C- or better is a passing grade.



©2019 Washington University in St. Louis
